Panel:
Whole System Virtualization in HEC Systems

Ron Brightwell

Scalable Computing Systems

Sandia National Laboratories

Albuquerque, New Mexico, USA

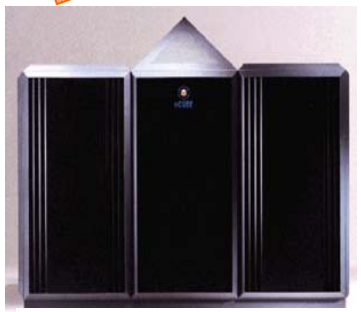
HPDC

July 26, 2005

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia Systems

1990



nCUBE2

- Sandia's first large MPP
- Achieved Gflop performance on applications

1993

Paragon

- 10s of users
- 1st periods processing MPP
- World record performance
- SUNMOS
- Routine 3D simulations

1997

ASCI Red

- Production MPP
- 100s of users
- Red & Black partitions
- Improved interconnect
- high-fidelity coupled 3d physics

1999

Cplant

- Commodity based supercomputer
- ~100's of users
- Linux-based OS licensed for commercialization
- Enhanced simulation capacity

2004

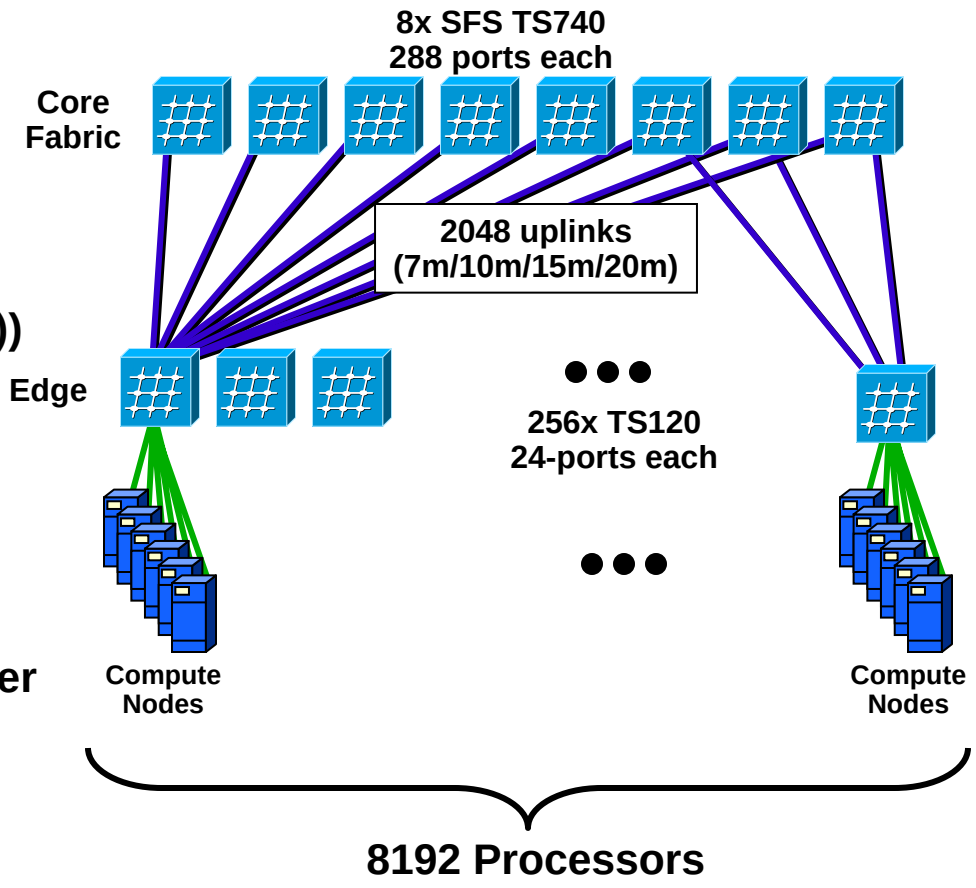
Red Storm

- ASCI's next flagship
- 41 Tflops
- Custom interconnect
- Purpose built RAS
- Highly balanced and scalable

Sandia National Laboratories

Sandia's 60 TF Thunderbird Machine

- Compute nodes
  - 4096 Dell Servers
  - Dual 3.6 GHz EM64T
  - 6 GB RAM
- Network
  - InfiniBand (Cisco (Topspin))
  - 50% Blocking Ratio
  - 8 TS-740s
  - 256 TS-120s
- Node count
  - Largest PC cluster in the world
  - Third largest Supercomputer in the world



National Nuclear Security Administration

Sandia National Laboratories

Characteristics of High-End Systems

- Applications are resource constrained – scaled to consume all of at least one resource
  - CPU, memory, memory bandwidth, network bandwidth, etc.
  - Applications manage the resources
- Machines are space-shared
  - Simple way to try to maximize resources
- Small set of devices
  - Compute nodes typically only have processors, memory, and network interfaces
- N = 1 (well, almost)

Sandia Lightweight Kernels (LWKs)

- Target high-performance scientific and engineering applications on tightly-coupled distributed-memory architectures
- Scalable to tens of thousands of processors
- Fast networking and execution
- Small memory footprint
- Persistent kernel

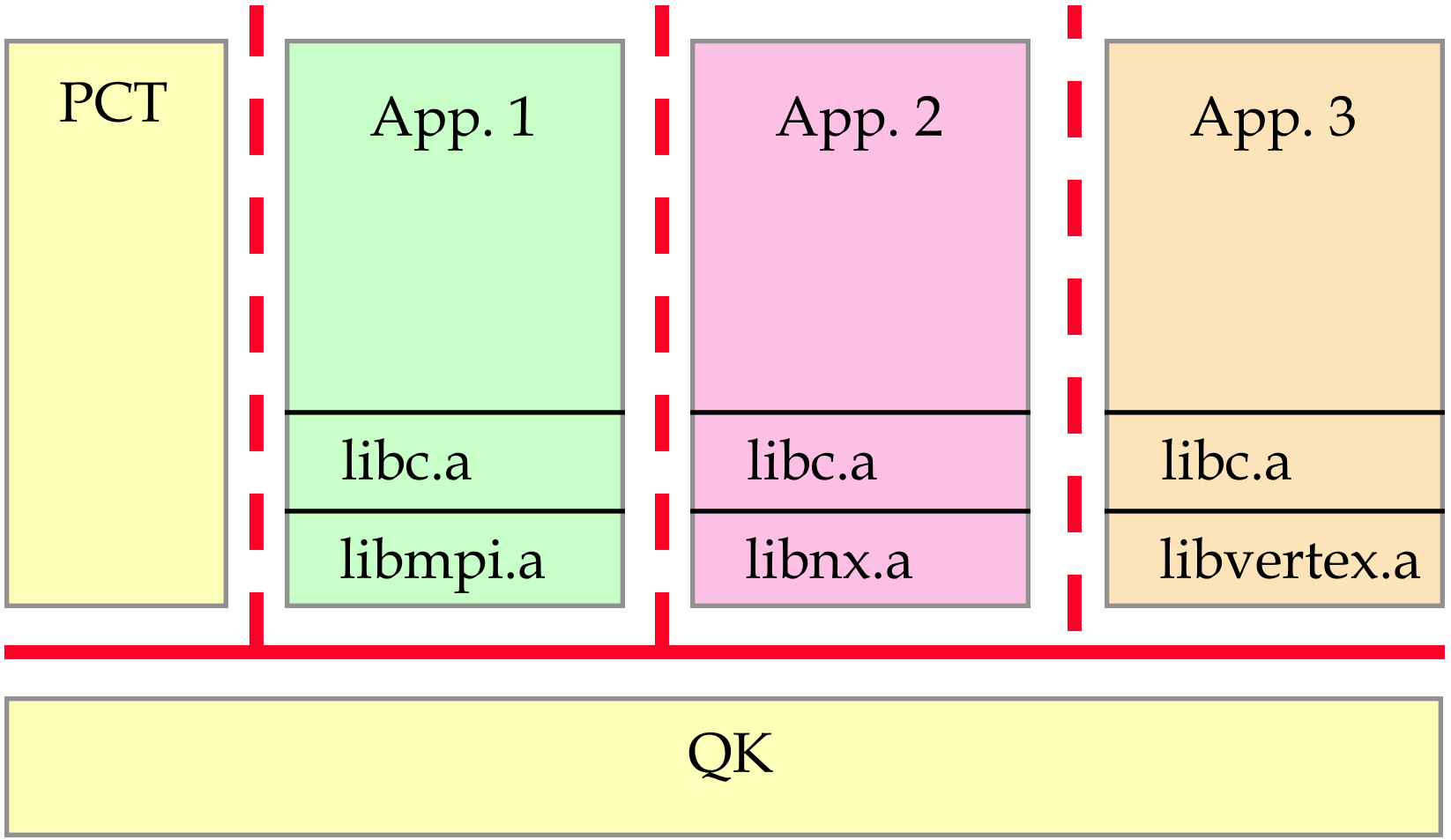
NNSA National Nuclear Security Administration

Sandia National Laboratories

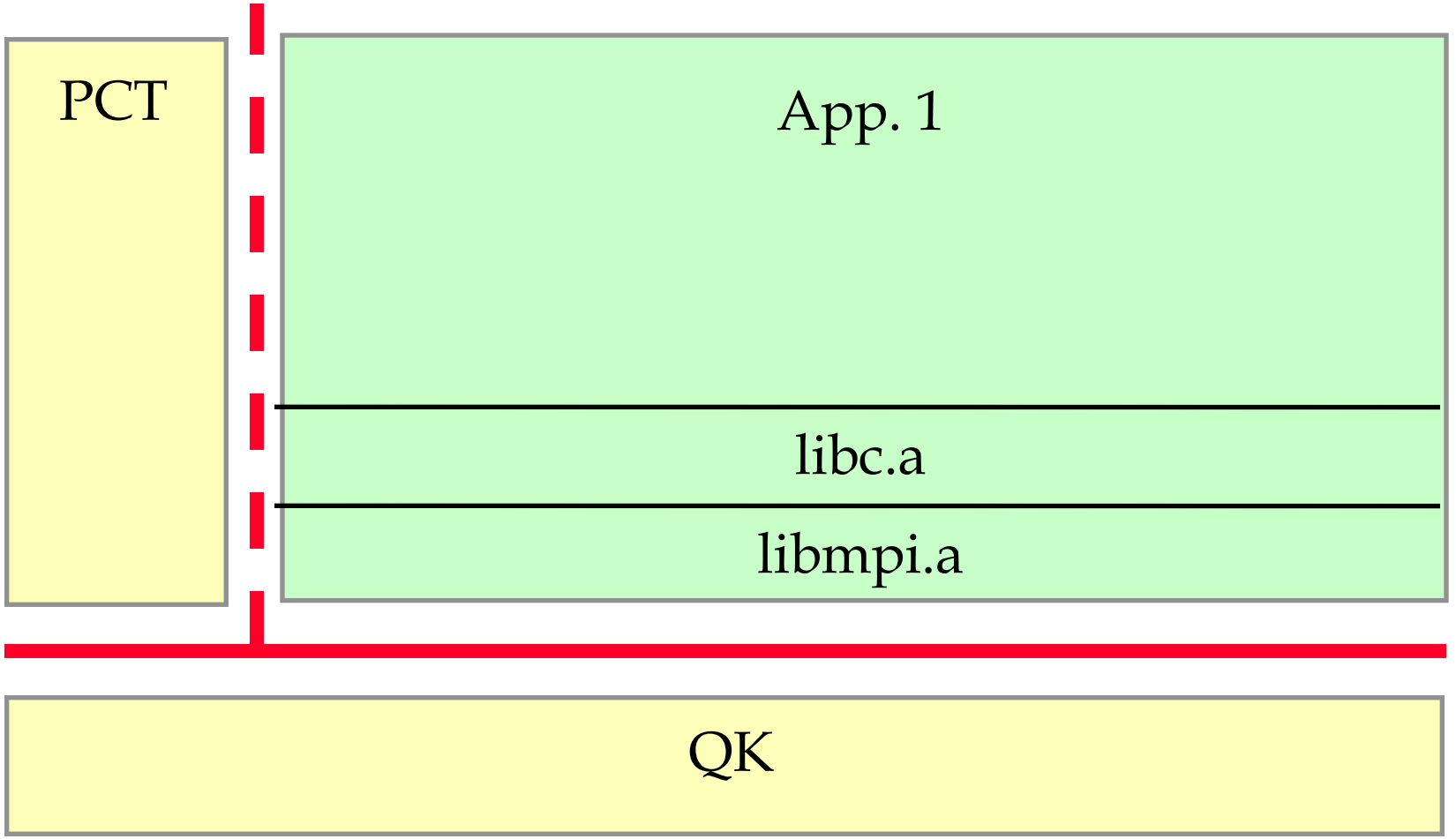
Approach

- Separate policy decision from policy enforcement
- Move resource management as close to application as possible
- Protect applications from each other
- Let user processes manage resources
- Get out of the way

LWK General Structure



Typical Usage



Quintessential Kernel (QK)

- Policy enforcer
- Initializes hardware
- Handles interrupts and exceptions
- Maintains hardware virtual addressing
- No virtual memory support
- Static size
- Small size
- Non-blocking
- Few, well defined entry points

NNSA National Nuclear Security Administration

Sandia National Laboratories

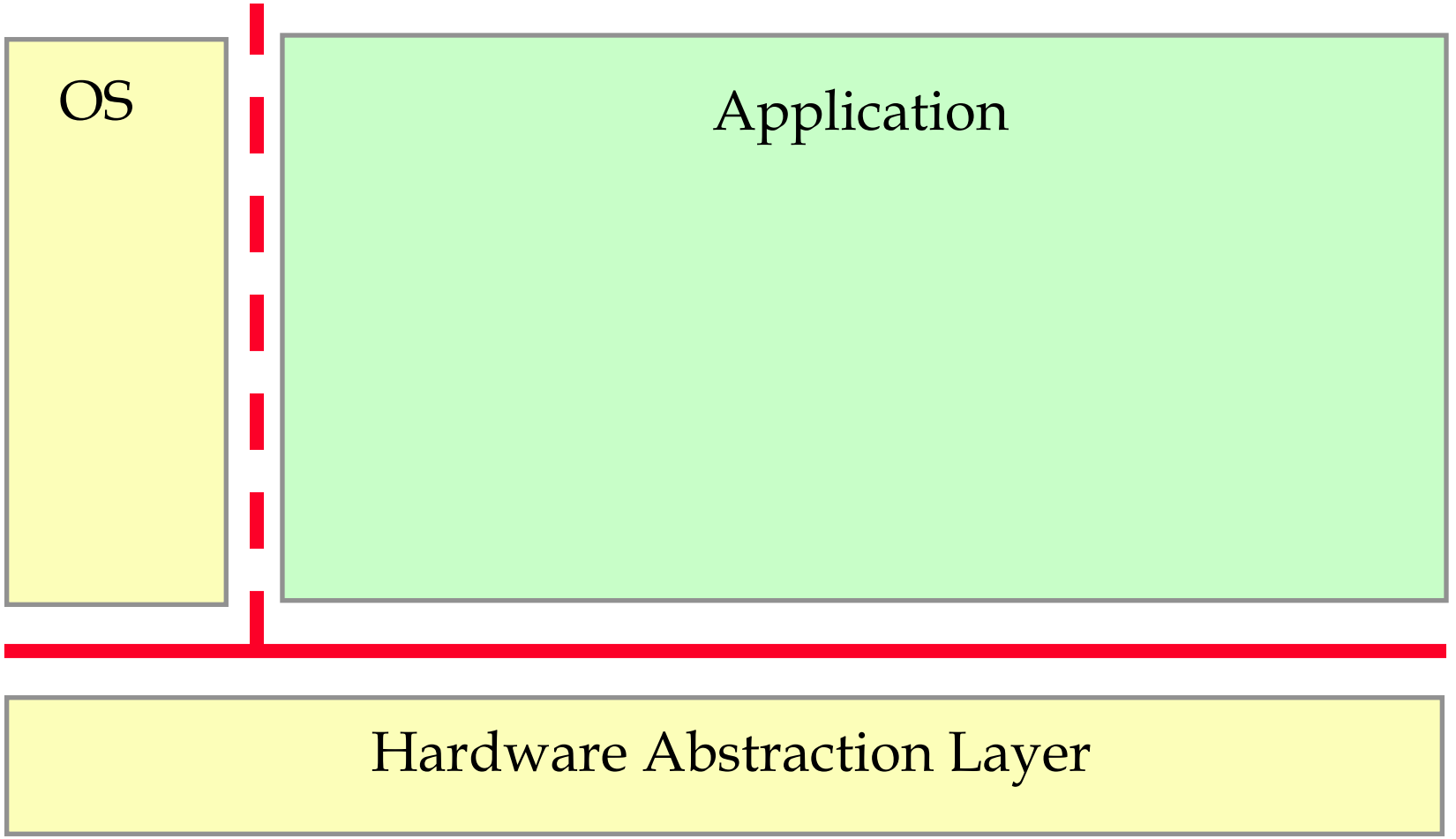
Process Control Thread (PCT)

- Runs in user space
- More privileged than user applications
- Policy maker
  - Process loading
  - Process scheduling
  - Virtual address space management
  - Name server
  - Fault handling

PCT (cont'd)

- Customizable
  - Singletasking or multitasking
  - Round robin or priority scheduling
  - High performance, debugging, or profiling version
- Changes behavior of OS without changing the kernel

Qk is Really a HAL



Current OS/Runtime Issues

- "OS Noise" or "Rogue OS effects"
  - LANL ASCI Q analysis
  - LLNL daemon scheduling
  - Upcoming workshop on OS interference
- Consistent page mappings for network devices
- Understanding and isolating the impact of the OS
- How much memory is there?
- The implementation and development of operating systems is an impediment to new architectures and programming models

Where Don't We Need Virtualization

- Processor
  - Can't leverage processor-specific features
    - i860 bus locking
- Memory
  - Linux already makes everything look like an x86
  - We already have enough problems with tracking memory usage
  - Applications always know better how to do resource allocation
- Network
  - No good way to provide isolation with network virtualization

Why We Might Need Virtualization

- OS development
  - Use VMM as hardware abstraction layer
    - No need to port to every new machine
  - Debugging
    - Easily capture entire state
  - Testbeds
- OS comparison
  - HAL makes direct OS performance comparison a little easier
    - Porting the OS isn't the issue – it was the network
- Checkpoint/Restart/Migration
  - For those who want this in the first place

All problems in computer science can be solved by another level of indirection.

-Butler Lampson

…except for performance.

-Me

National Nuclear Security Administration

Sandia National Laboratories